1   1.   A method for accessing a storage system comprising:

2   accessing a data object, the data object being divisible into one or more

3   partitions, the partitions comprising data from the data object, the partitions referred to as

4   input partitions; and

5   for each input partition, if there are no other partitions in the storage system

6   that are identical to the input partition, then producing one or more replicas of the input

7   partition.

1   2.   The method of claim 1 wherein the data object is a file.

1   3.   The method of claim 1 wherein a first partition is associated with a

2   partition ID, such that another partition having content that is identical to content of the first

3   partition, then the other partition is associated with the same partition ID as that of the first

4   partition, the method further comprising storing partition identification information

5   comprising a plurality of partition IDs, each partition ID being associated with one or more

6   partitions, determining a partition ID of the input partition, accessing the partition identity

7   information to determine if there are any partitions that are identical to the input partition

8   based on the partition ID of the input partition.

1   4.   The method of claim 1 wherein the data object is to be stored on the

2   storage system, the method further comprising receiving a request to store the data object on

3   the storage system, receiving data comprising the data object, and storing the data object on

4   the storage system.

1   5.   The method of claim 7 further comprising, for each input partition,

2   generating a content-based identifier based on at least some content of the input partition and

3   identifying first partitions in the storage system that have the same content-based identifier,

4   wherein the one or more replicas are produced if none of the first partitions is identical to the

5   input partition.

1   6.   The method of claim 5 wherein the step of generating includes

2   applying a hash algorithm to at least a portion of the content of the input partition.

1   7.   The method of claim 7 wherein the data object is a file.

8.   A method for accessing a storage system comprising:

    receiving data for a first file, to be stored in the storage system;

    providing partition data from the first file which constitutes a first partition of the first file;

    if a number of second partitions in the storage system is less than a first predetermined value, then producing a number of replicas of the first partition sufficient to increase the number of second partitions to a second predetermined value, wherein each second partition comprises data that is identical to the partition data; and

    if the number of second partitions is greater than a third predetermined value and if there are one or more replicas of the first partition, then deleting one or more of the replicas, wherein the number of second partitions is reduced; and

    repeating for additional partition data comprising the first file.

9.   The method of claim 8 further comprising receiving a request to store the first file, receiving data comprising the first file, and storing the first file on the storage system.

10.   The method of claim 8 wherein partitions each is identified by a content-based code and a group ID, wherein if data one partition is different from data of another partition and both partitions have the same content-based code, then each partition has a different group ID, whereby partitions that contain identical data are identified by the same content-based code and group ID.

11.   The method of claim 10 further comprising storing partition identity information comprising a content-based code and a group ID that correspond to each partition on the storage system, wherein the first partition is associated with a first content-based code value and a first group ID value, wherein the number of second partitions can be determined by consulting the partition identity information and counting the number of partitions whose corresponding content-based code is equal to the first content-based code value and whose corresponding group ID is equal to the first group ID value.

12.   The method of claim 8 wherein the content-based code is a hash code produced by applying a hash algorithm to content of a partition.

20

1   13.   The method of claim 10 further comprising storing partition identity

2   information comprising a hash code and a group ID that correspond to each partition on the

3   storage system, wherein the first partition is associated with a first hash code value and a first

4   group ID value, wherein the step of producing a number of replicas includes adding

5   information which identifies each replica to the partition identity information, including the

6   first hash code value and the first group ID value.

1   14.   The method of claim 8 wherein the first predetermined value is less

2   than the second predetermined value.

1   15.   The method of claim 8 wherein the first predetermined value is equal

2   to the second predetermined value.

1   16.   The method of claim 8 wherein the step of deleting one or more

2   replicas further includes deleting one or more replicas until all the replicas are deleted or until

3   the number of second partitions is less than a fourth predetermined value.

1   17.   The method of claim 16 wherein the third predetermined value is

2   greater than the fourth predetermined value.

1   18.   A method for accessing a storage system comprising:

2   receiving a request to store a file;

3   storing the file on the storage system;

4   identifying one or more partitions which collectively constitute the file, the

5   partitions referred to as input partitions;

6   storing partition information that is associated with the file, wherein the

7   partition information associates the file with each of its input partitions; and

8   for each input partition, if there are no identical partitions, then

9   if the number of replicas of the input partition is less than a threshold

10   value, then producing at least one replica of the input partition and storing the replica

11   on the storage system,

12   wherein an identical partition is a partition, other than the input

13   partition, of a file that is stored in the storage system whose content is identical to

14   content of the input partition.

1   19.   The method of claim 18 wherein for a first input partition, if there is at

2   least one other file that comprises a partition that is identical to the first input partition, then

3   deleting a replica of the first input partition if the replica exists.

1   20.   The method of claim 18 wherein for an input partition, the number of

2   identical partitions plus the number of replicas is equal to a predetermined value, the

3   threshold value being the difference between the predetermined value and the number of

4   identical partitions.

1   21.   The method of claim 18 wherein the partition information associated

2   with the file comprises a partition ID for each input partition, wherein the partition ID

3   comprises a hash code and a group ID, wherein the hash code is determined by applying a

4   hash function to contents of a partition, wherein the group ID identifies a partition whose

5   content is unique among partitions which have the same hash code.

1   22.   The method of claim 18 further comprising storing information that

2   identifies one or partition groups, a partition group comprising one or more partitions

3   identified from among one or more files which contain identical content, a partition group

4   further comprising one or more replicas of a partition in the partition group.

1   23.   The method of claim 18 wherein the threshold value is one.

1   24.   The method of claim 18 wherein the threshold value is a number

2   greater than one.

1   25.   The method of claim 18 wherein each partition is the same size as

2   other partitions.

1   26.   The method of claim 18 wherein identifying one or more partitions

2   includes determining a partition size by which the partitions of the file are identified.

1   27.   The method of claim 18 wherein a partition size of partitions of a file

2   can be different for different files.

1   28.   A data storage system comprising:

2   a storage component; and

3   a data processing component in data communication with the storage

4 component, the data processing component for receiving access requests from users, the

5 access requests for accessing data that is stored in the storage component or for storing data

6 to the storage component,

7   the data processing component configured to perform the method steps of:

8   accessing a first partition of a file, the first partition comprising a first

9 portion of data that constitutes the file;

10   if the first partition does not have a corresponding identical partition in

11 the storage component, then creating at least one replica; and

12   repeating for a second partition of the file, the second partition

13 comprising a second portion of the data.

1   29.   The system of claim 28 wherein the data processing component is

2 further configured to perform the method steps of:

3   receiving a request to store data to the storage system, the request including

4 the data that constitutes the file; and

5   accessing the storage component to store the data.

1   30.   The system of claim 28 wherein a partition is identified with a partition

2 ID, the partition ID being based on content of the partition, wherein partitions which contain

3 identical content have the same partition ID,

4   the storage system further comprising information which is stored in the

5 storage component, each partition having its associated partition identity information, the

6 partition identity information comprising a partition ID its associated partition, wherein

7 partitions that are identical have the same partition ID,

8   whereby identical partitions can be identified by consulting the partition

9 identity information.

1   31.   The system of claim 30 wherein the partition ID comprises a content-

2 based code and a group ID, wherein the content-based code is determined from the content of

3 a partition, wherein if one partition and another partition have the same content-based code

4 but have different content, then the one partition is associated with a first group ID and the

23

5   other partition is associated with a second group ID different from the first group ID, wherein

6   if the one partition and the other partition have the same content-based code and have

7   identical content, then the one partition and the other partition both are associated with the

8   same group ID.

1   32.   The system of claim 31 wherein the content-based code is a hash code,

2   wherein the hash code is generated by applying a hash function to the content of a partition.

1   33.   A data processing system comprising:

2   first means for producing a partition ID for each partition comprising a file, a

3   partition comprising data from the file, the first means producing a first partition ID for a first

4   partition of a first file;

5   second means for identifying one or more identical partitions in the storage

6   system based on a first partition ID; and

7   third means for creating a replica of the first partition in response to the second

8   means making a determination that there are no identical partitions,

9   wherein the first means, the second means, and the third means operate on

10   every partition comprising the first file.

1   34.   The system of claim 33 wherein a partition ID comprises a hash code

2   and a group ID, the first means comprising hash means for producing a hash code based on

3   data comprising a partition and group ID means for determining a group ID, wherein

1   35.   A method for accessing a storage system comprising:

2   accessing a first read-out partition of a data object, the first read-out partition

3   comprising at least a portion of content comprising the data object;

4   if content in the first read-out partition is corrupted, then:

5   accessing the storage system to find a replacement partition from

6   among one or more candidate partitions, including determining if a candidate partition

7   is corrupted or not, the replacement partition being a candidate partition that is not

8   corrupted; and

9   replacing content in the data object that constitutes the first read-out

10   partition with content of the replacement partition; and

11   repeating the foregoing with a second read-out partition of the data object.

1 36. The method of claim 35 wherein each partition is associated with a
2 partition ID, wherein if two partitions comprise identical content, then the two partitions have
3 the same partition ID, wherein the read-out partition is associated with a first partition ID and
4 the one or more candidate partitions each is associated with the first partition ID.

1 37. The method of claim 36 wherein the partition ID comprises a hash
2 code and a group ID, wherein the hash code is based on content of a partition, wherein if the
3 two partitions have the same hash code but their respective content is different from each
4 other, then the two partitions each have a different group ID, wherein if the two partitions
5 comprise identical content, then the two partitions have the same group ID.

1 38. The method of claim 35 wherein if a replacement partition cannot be
2 found for the first read-out partition, then indicating an error condition.

1 39. The method of claim 35 further comprising receiving a data access
2 request, the data access request including information indicative of the data object.

1 40. The method of claim 35 wherein the data object is a file.

1 41. A method for accessing data in a storage system comprising:
2   identifying a first data object;
3   obtaining a first partition of the first data object from the storage system;
4   performing a computation using data comprising the first data object to
5 produce a first computed value;
6   obtaining partition identification information relating to the first partition, the
7 partition identification information including a first previously computed value; and
8   if the first computed value does not match the first previously computed value,
9 then:
10   obtaining a first candidate partition from the storage system;
11   performing a computation using data comprising the first candidate
12   partition to produce a second computed value;
13   obtaining partition identification information relating to the first
14   candidate partition, the partition identification information including a second
15   previously computed value;

16   if the second computed value does not match the second previously

17   computed value, then repeating with a second candidate partition; and

18   if the second computed value does match the second previously

19   computed value, then replacing the data comprising the first partition of the first data

20   object with the data comprising the first candidate partition.

1   42.   The method of claim 41 wherein identifying a first data object is a step

2   of receiving a read request for the first data object.

1   43.   The method of claim 41 wherein if the first computed value does

2   match the first previously computed value, then repeating with a second partition of the first

3   data object.

1   44.   The method of claim 41 wherein the first data object is a file.

1   45.   The method of claim 41 wherein steps comprising the method are

2   repeated for a second data object.

1   46.   A data storage system comprising:

2   a storage subsystem; and

3   a data processing subsystem in data communication with the storage

4   subsystem to store data to the storage subsystem and to access data stored on the storage

5   subsystem,

6   the data processing subsystem configured to:

7   access a first file stored on the storage subsystem, wherein the first file

8   comprises data, the data being logically grouped into one or more accessed partitions;

9   determine, for each accessed partition, whether the accessed partition

10   is corrupt, referred to as a corrupt partition;

11   determine, for each corrupt partition, whether there is a replacement

12   partition on the storage system, the replacement partition being identical to the

13   accessed partition at a time when the accessed partition was not corrupt; and

14   modify the first file to replace each of its corrupt partitions with a

15   replacement partition if it exists.

1   47.   The system of claim 46 wherein the data processing subsystem is

2   further configured to communicate with one or more users and to receive access requests for

3   data stored on the storage subsystem, wherein one such access request is a request for the first

4   file.

1   48.   The system of claim 46 further comprising partition identity

2   information, wherein each partition on the storage subsystem is associated with its

3   corresponding partition identity information, the partition identity information including a

4   partition ID, wherein the partition ID uniquely identifies content of a partition, wherein

5   partitions that are identical have the same partition ID,

6   wherein the data processing subsystem is further configured to determine

7   whether an accessed partition is corrupt based on its accessed content and on its partition ID.

1   49.   The system of claim 46 wherein each partition on the storage

2   subsystem is associated with a partition ID comprising a hash code component and a group

3   ID component, wherein partitions that have the same hash code value also have the same

4   group ID if they have identical content, wherein the data processing subsystem is further

5   configured to:

6   compute a first hash value for an accessed partition; and

7   compare the first hash value with the hash code component of the partition ID

8   associate with the accessed partition in order to determine whether the accessed partition is

9   corrupt.

1   50.   A data processing system comprising:

2   first means for accessing a partition comprising a file that is stored on a

3   storage subsystem, a partition comprising data from the file, a partition having associated

4   therewith a partition ID that uniquely identifies content of the partition, wherein partitions

5   comprising identical content have the same partition ID;

6   second means for determining whether a partition is corrupt;

7   third means for identifying a replacement partition from among a plurality of

8   partitions stored on the storage subsystem to replace a corrupt partition, based on a partition

9   ID associated with the corrupt partition and on partition IDs of the plurality of partitions, the

10   corrupt partition being a constituent partition of a target file; and

11   fourth means to modify the target file to replace content comprising the

12   corrupt partition with content from a replacement partition.